PB-212 625

OFF-LINE ANALYSIS PROGRAMS FOR LONG-PATH SPECTROMETER

Roger H. Hansen, et al

Bendix Aerospace Systems Division

Prepared for:

Environmental Protection Agency

15 June 1972

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

EPA-R2-72-092

PB 212 625

Off-Line Analysis Programs For Long-Path Spectrometer

Final Report
BSR 3322

Ann Arbor, Michigan

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

Bendix Aerospace Systems Division

57

BIBLIOGRAPHIC DATA SHEET	1. Report No. EPA-R2-72-092	2.	3. Recipient's Accession No. PB-212 625
4. Title and Subtitle: Off-Line Analysis Programs for Long-Path Spectrometer			5. Report Date: June 15, 1972
			6.
7. Author(s): Roger H. Hansen and Robert H. Dye			8. Performing Organization Rept. No. BSR 3322
9. Performing Organization Name and Address: BENDIX, Aerospace Systems Division, Ann Arbor, Michigan			10. Project/Task/Work Unit No.
			11. Contract/Grant No. 68-02-0249
12. Sponsoring Organization Name and Address: ENVIRONMENTAL PROTECTION AGENCY, Research Triangle Park, North Carolina 27711			13. Type of Report & Period Covered
			14.

15. Supplementary Notes

16. Abstracts

The report documents thirteen Off-Line Analysis Programs for the Long-Path Infrared Spectrometer instrument and provides user instructions for implementing the programs on an IBM-360/50 computer. Theoretical discussion is included only to the extent necessary to permit the proper use of the programs.

17. Key Words and Document Analysis. 17a. Descriptors

Infrared spectrophotometers
Absorptivity
Programs (computers)
Routines (computers)
Instructions
Computers
Spectra
Transmittance
Regression analysis
Factor analysis
Standard deviation

17b. Identifiers/Open-Ended Terms

Long path spectrometers
Ozone Monitor Instruments

17c. COSATI Field/Group 13B, 20F, 9B

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Unlimited	UNCLASSIFIED	57
	20. Security Class (This Page) UNCLASSIFIED	22. Price $3.00

FORM NTIS-35 (REV. 3-72)

THIS FORM MAY BE REPRODUCED

USCOMM-DC 14952-P7

Off-Line Analysis Programs For Long-Path Spectrometer

Final Report
BSR 3322

Ann Arbor, Michigan

Prepared for:
Environmental Protection Agency
Research Triangle Park,
North Carolina

By:
Roger H. Hansen
Robert H. Dye

Contract No. : 68-02-0249

Bendix Aerospace Systems Division

II

CONTENTS

		Page
1.	INTRODUCTION	1-1
2.	DOCUMENTATION	2-1
	2.1 HEXED PROGRAM	2-1
	2.2 OMAS PROGRAM	2-3
	2.3 EPASTREG PROGRAM	2-5
	2.4 OMCC PROGRAM	2-27
	2.5 EPAMSTDV PROGRAM	2-29
	2.6 EPAPLOTR PROGRAM	2-30
	2.7 FACTORAN PROGRAM	2-31
	2.8 FACREC PROGRAM	2-37
	2.9 DIMAT PROGRAM	2-39
	2.10 ALFMAT PROGRAM	2-42
	2.11 FILTER PROGRAM	2-43
	2.12 DICIS PROGRAM	2-45
	2.13 PLTSMF PROGRAM	2-50
3.	REFERENCES	3-1

SECTION 1

INTRODUCTION

The purpose of this report is to document the Off-Line Analysis Programs for the Long-Path Infrared Spectrometer instrument and to provide user instructions for implementing the programs on an IBM-360/50 computer. Theoretical discussion is included only to the extent necessary to permit the proper use of the programs.

Thirteen computer programs are documented. They are listed below by number and name:

1. HEXED - Hexidecimal Edit Program
2. OMAS - Ozone Monitor Absorption Simulator Program
3. EPASTREG - EPA Stepwise Regression Program
4. OMCC - Ozone Monitor Coefficient Converter Program
5. EPAMSTDV - EPA Mean and Standard Deviation Program
6. EPAPLOTR - EPA Plotter Program
7. FACTORAN - Factor Analysis Program
8. FACREC - Factor Recombination Program
9. DIMAT - Deconvolution and Interpolation Matrix Program
10. ALFMAT - Absorption Contaminant Matrix Program
11. FILTER - Spectral Filter Data Program
12. DICIS - Deconvolution, Interpolation, Contamination, and Instrument Simulation Program
13. PLTSMF - Plot Spectral Modulation Functions Program.

SECTION 2

DOCUMENTATION

2.1 HEXED PROGRAM

2.1.1 Purpose

This program edits spectral data obtained from the Long Path Spectrometer (LPS) in the form of a four hexadecimal digit number for each channel reading. The data are first processed through paper-tape-to-card conversion equipment to produce data cards which are the input media for this program. A complete set of channel readings, representing one spectral scan, is accompanied by an ID number and a checksum. The program recomputes a checksum for the data and a comparison is made with the checksum associated with the data set. If the comparison fails, the data are rejected.

The channel readings are converted from hexadecimal-coded numbers to decimal numbers in the program, and those data sets which pass the checksum comparison are punched on an output card file for subsequent use in numerical computations.

Through the use of control options, the program will list all the data cards as read, list only those card sets in error, list the converted card images, list the filter wheel position list, and punch the filter wheel position list on the output card file.

2.1.2 Control Cards: Fortran Unit 5

1. Columns 1-4: One of the following options: (one to five option cards)

   a. PRNT - Print filter wheel position list.
   b. FWPL - Write filter wheel position list on output file.
   c. CDIS - List contents of input records as read.
   d. CVRF - List converted input records.
   e. EXEC - Begin program execution (must be last option card).

   Columns 5-80: Blank.

2. DAT (I), I=1, NOC+2

DAT (1) = Line Feed code (LF=Z69)
DAT (I), I = 2, 5 = NOC (number of channels)
DAT (I), I = 6, NOC, 4 (I<66) = Filter wheel position list
DAT (66) = Carriage return code (CR=Z7A)

3. Cards 3 - (2 + NOC/16)

DAT (1) = LF
DAT (I), I = 2, NOC, 4 (I<66) = Filter wheel position list
DAT (66) = CR
DAT (I), I = NOC + 1 = Checksum
DAT (I), I = NOC + 2 = CR

2.1.3 Data Input: Fortran Unit 5

Following are the scan data cards from the LPS:

DAT (1) = LF
DAT (I), I = 2, NOC, 4 (I<66) = Channel readings
DAT (66) = CR
DAT (NOC + 1) = ID number
DAT (NOC + 2) = Checksum
DAT (NOC + 3) = CR

2.1.4 Output Data File: Fortran Unit 7

This unit produces the punched-card output for input to OMAS:

DAT (1) = NC = Number of cases
DAT (2) = NOC = Number of channels
DAT (I), I = 3, NOC + 3, 4 = Filter wheel position list (FWPL. TRUE.)
DAT (I), I = NOC + 4, 2* NOC + 4, 4 = Channel readings
DAT (2* NOC + 5) = ID number

2.1.5 Temporary Data File: Fortran Unit 8

2.1.6 Printer: Fortran Unit 6

2.2 OMAS PROGRAM

2.2.1 Purpose

This program computes the transmittance that would be in each spectral channel if there had been an amount of ozone in the beam giving a specified peak fractional absorptance. This computed transmittance takes into account the spectral bandwidth of the Ozone Monitor and is needed to simulate ozone-containing spectra.

2.2.2 Control Cards: Fortran Unit 1

1. NOP, NOD, NOAPP, NTSC; Format (I10) - Four cards

   NOP - Number of problems (contaminants) = 1 for Ozone Monitor

   NOD - Number of channels deleted from computation = 0 for Ozone Monitor

   NOAPP - Number of agents per problem = 1 for Ozone Monitor

   NTSC - Number of spectral scans to skip

2. AAPK, NAM, PRINTF, BW; Format (8G10.0) - One card

   AAPK - Agent absorption peak

   NAM - Background-to-Agent case ratio = 1 for Ozone Monitor

   PRINTF - Print flag (1 = print; 0 = no print)

   BW - Bandwidth of spectral channels = 0.15 for Ozone Monitor

3. NAGENT, NC, (ALPHA (I), I = 1, 19); Format (2A4, I2, 7E10.0)

   NAGENT - Name of contaminating agent

   NC - Card number

   ALPHA(I) - Absorption coefficients for agent (7 per card)

2.2.3 Data Input: Fortran Unit 4 (reads HEXED punched output cards)

1. N; Format (G10.0) (one card)

   N - Number of cases

2. NOC, FWPOS(I), I = 1, NOC; Format (8G10.0)

   NOC - Number of channels

   FWPOS(I) - Filter wheel position list

3. RAD(J), J = 1, NOC, IDN; Format (20A4)

   RAD(J) - Channel readings for one spectral scan

   IDN - ID number for scan group

2.2.4 Data Output File: Fortran Unit 5 (card punch output for EPASTREG)

1. NCS, NCS; Format (2I10)

   NCS - Number of output cases

2. RAD(I), IDN; Format (6F10.5, 10X, I10)

   RAD(I) - Convolved output data

   IDN - ID number for scan group

2.2.5 Printer: Fortran Units 3 and 6

2.3 EPASTREG PROGRAM

This program is a modification of a regression analysis program produced by the Health Science Computing Facility of the University of California. For a further discussion of regression analysis, see Ref. 1*.

2.3.1 General Description

This program computes a sequence of multiple linear regression equations in a stepwise manner. At each step, one variable is added to the regression equation. The variable added is the one which makes the greatest reduction in the error sum of squares. Equivalently it is the variable which has highest partial correlation with the dependent variable partialed on the variables which have already been added; and equivalently it is the variable which, if it were added, would have the highest F value. In addition, variables can be forced into the regression equation and automatically removed when their F values become too low. Regression equations with or without the regression intercept may be selected.

The output from this program includes:

(1) At each step:

(a) Multiple R

(b) Standard error estimate

(c) Analysis-of-variance table

(d) For variables in the equation:

1. Regression coefficient

2. Standard error

3. F to remove

(e) For variables not in the equation:

*See Section 3 for references.

1. Tolerance

2. Partial correlation coefficient

3. F to enter

(2) Optional output prior to performing regression:

(f) Means and standard deviations

(g) Covariance matrix

(h) Correlation matrix

(3) Optional output after performing regression:

(i) List of residuals

The limitations per problem are:

(1) p, number of original variables ($2 \leq p \leq 141$)

(2) q, number of variables added by transgeneration ($-9 \leq q \leq 78$)

(3) p+q, total number of variables ($2 \leq p+q \leq 141$)

(4) s, number of Subproblem Cards ($1 \leq s \leq 99$)

(5) k, number of Variable Format Cards ($1 \leq k \leq 10$)

(6) i, number of variables to be plotted ($0 \leq i \leq 30$)

(7) m, number of Transgeneration Cards ($0 \leq m \leq 99$)

This program allows transgeneration of the variables. Codes 01-17 and 20-24 of the transgeneration list may be used.

2.3.2 Order of Cards in Job Deck

Cards indicated by letters enclosed in parentheses are optional. All other cards must be included in the order shown:

a. System Cards

b. Problem Card

(c.) Transgeneration Card(s)

(d.) Labels Card(s)

e. DATA INPUT Cards
(Place data input deck here if data input is from cards.)

f. Subproblem Card(s)

g. Control-Delete Card(s)

h. through (i.) may be repeated as many as 99 times in each problem; b. through (i.) may be repeated as often as desired

i. Finish Card

2.3.3 Card Preparation (Specific for This Program)

Preparation of the cards listed below is specific for this program. All other cards listed in the preceding section are prepared according to instructions in the Introduction.

b. Problem Card (one for each problem)

Col. 1-6	BLANK	(Mandatory)
Col. 8-15	Alphanumeric problem name	
Col. 17-20	Sample size ($1 \leq n \leq 9999$)	

Co. 23-25	Number of original variables ($2 \leq p \leq 141$) (Dependent plus independent variables).
Col. 28-30	Number of Transgeneration Cards ($0 \leq m \leq 99$).
Col. 33-35	Number of variables added by transgeneration ($-9 \leq q \leq 78$).
Col. 43-45	Number of Subproblem Cards ($1 \leq s \leq 99$).
Col. 47-49	Number of variables labeled on Labels Cards. Leave blank if Lables Cards are not used.
Col. 51-53	1 if means and standard deviations are to be printed; otherwise, leave blank.
Col. 55-57	1 if covariance matrix is to be printed; otherwise, leave blank.
Col. 59-61	1 if correlation matrix is to be printed; otherwise, leave blank.
Col. 63-65	1 if zero regression intercept is desired; otherwise, leave blank.
Col. 68-69	1 if input tape to be rewound.
Col. 72	1 if Matrix to be printed at each step.
Col. 73	1 if input data to be printed.
Col. 74	1 if ID number changes to be printed (must be blank if variable format is used).
Col. 75	The number of variable format cards. (If blank program assumes ID input scheme to be used).

g. Subproblem Card

Col. 1-6	333333	(Mandatory).
Col. 9, 10	Number of the dependent variable.	
Col. 14, 15	Maximum number of steps. This will be 2(p+q) if left blank.	
Col. 21-30	F-level for inclusion. This will be 0.01 if left blank.	
Col. 31-40	F-level for deletion. This will be 0.005 if left blank.	
Col. 41-50	Tolerance level. This will be 10^{-8} if left blank.	
Col. 53-55	1 if Control-Delete Cards are included.	
Col. 58-60	1 if list of residuals is to be printed.	
Col. 65	1 if Retreat mode to be used.	

(h.) Control-Delete Card

Col. 1-6	444444	(Mandatory).
Col. 7	Control value* for first variable.	
Col. 8	Control value* for second variable	
	...	
Col. 72	Control value* for 66th variable.	

(i.) Labels Cards

Labels Cards allow the user to substitute alphanumeric names for the usual numeric indexes (variable numbers or category designations) which appear on the printed output.

Card Preparation

Col. 1-6	LABELS	(Mandatory).
Col. 7-10	The number of the variable (or category or index) to be named. This number must be right-justified.	
Col. 11-14	The corresponding alphanumeric name.	
Col. 17-20	The number of another variable.	
Col. 21-24	The corresponding alphanumeric name.	
Col. 67-70	The number of another variable.	
Col. 71-76	The corresponding alphanumeric name of that variable (up to 7 per card).	

There may be from one to seven pairs of variable numbers and labels on each Labels Card. If desired, only one pair may be specified on each card. However, the total number of labels appearing on all the Labels Cards must equal the number of labels specified on the Problem or Subproblem Card. It is not necessary to label all the variables. Those labeled may be listed in any order.

Example: Suppose the number of variables to be labeled as specified on the Problem Card is 9. Then the Labels Cards might be punched as: LABELS 10HEIGHT
07WEIGHT 105AGE 003X1 0051VAR59 0073 X+Y LABELS
99SE 0100ANYNAM
LABELS 05STATUS

(j.) Transgeneration Cards

The term transgeneration is used to include transformations of input variables and creation of new variables prior to the normal computations performed by the various programs. The transformations described below are

performed on the values of the variables in each case. In these examples, the symbol x_i will denote the i^{th} variable as well as its value.

Examples:

$\log_{10} X_4 \rightarrow X_4$ $\log_{10} X_4$ replaces X_4

$X_5^c \rightarrow X_1$ X_5^c replaces X_1

$X_2 + X_3 \rightarrow X_2$ $X_2 + X_3$ replaces X_2

By successive transformations, more complicated relationships may be obtained. For example:

(1) To replace X_5 by $\sqrt{X_1^2 + X_3^2}$ four transformations are required:

Transformation	Variables as they are stored at each step				
	X_1	X_2	X_3	X_4	X_5
$X_1^2 \rightarrow X_1$	X_1^2	X_2	X_3	X_4	X_5
$X_3^2 \rightarrow X_3$	X_1^2	X_2	X_3^2	X_4	X_5
$X_1 + X_3 \rightarrow X_5$	X_1^2	X_2	X_3^2	X_4	$X_1^2 + X_3^2$
$\sqrt{X_5} \rightarrow X_5$	X_1^2	X_2	X_3^2	X_4	$\sqrt{X_1^2 + X_3^2}$

In this example, it can be seen that the original values of X_5 are irrelevant. Actually, the variable X_5 may be a dummy variable introduced by the program specifically to provide capacity for creating new variables by transgeneration. Dummy variables may be required for intermediate storage to effect some transformations.

(2) To replace X_1 by $\exp(-1/2\ X_1^2)$, three transformations are required:

Transformation			X_1	X_2	X_3
X_1^2	→	X_1	X_1^2	X_2	X_3
$-1/2\ X_1$	→	X_1	$-1/2\ X_1^2$	X_2	X_3
$\exp(X_1)$	→	X_1	$\exp(-1/2\ X_1^2)$	X_2	X_3

(3) To replace X_4 by $X_2 + \log_{10}(X_4 - X_3 + 100)$, four transformations are required:

Transformation			X_1	X_2	X_3	X_4
$X_4 - X_3$	→	X_4	X_1	X_2	X_3	$X_4 - X_3$
$X_4 + 100$	→	X_4	X_1	X_2	X_3	$(X_4 - X_3 + 100)$
$\log_{10} X_4$	→	X_4	X_1	X_2	X_3	$\log_{10}(X_4 - X_3 + 100)$
$X_2 + X_4$	→	X_4	X_1	X_2	X_3	$X_2 + \log_{10}(X_4 - X_3 + 100)$

The transformations are performed in the order in which the Transgeneration Cards appear so that, for example, the two transgenerations $2X_1 \rightarrow X_1$ followed by $X_1 - 2 \rightarrow X_1$ will result in $2X_1 - 2$, whereas $X_1 - 2 \rightarrow X_1$ followed by $2X_1 \rightarrow X_1$ will result in $2(X_1 - 2)$.

2.3.3.1 Transgeneration List

The notation to be used in the following transgeneration list is as follows:

1. i, j, k are variable indexes (need not be different)

2. c is a constant

3. a_1, a_2, a_3, ... are constants

4. n is the number of cases, or sample size

5. The mean $\overline{X}_i = \frac{1}{n} \sum_{n=1}^{n} X_{ji}$

6. The standard deviation $s_i = \left[\frac{1}{n-1} \sum_{j=1}^{n} (X_{ji} - \overline{X}_i)^2 \right]^{1/2}$

Code	Transgeneration	Restriction
01	$\sqrt{X_i} \rightarrow X_k$	$X_i \geq 0$
02	$\sqrt{X_i} + \sqrt{X_i + 1} \rightarrow X_k$	$X_i \geq 0$
03	$\log_{10} X_i \quad X_k$	$X_i > 0$
04	$e^{X_i} \rightarrow X_k$	-
05	$\arcsin \sqrt{X_i} \rightarrow X_k$	$0 \leq X_i \leq 1$
06	$\arcsin \sqrt{X_i / (n+1)}$ + $\arcsin \sqrt{(X_i + 1) / (n+1)} \rightarrow X_k$	$0 \leq (X_i \; n) \leq 1$
07	$1/X_i \rightarrow X_k$	$X_i \neq 0$
08	$X_i + c \rightarrow X_k$	-
09	$X_i c \rightarrow X_k$	-

Code	Transgeneration	Restriction
10	$X_i^c \rightarrow X_k$	-
11	$X_i + X_j \rightarrow X_k$	-
12	$X_i - X_j \rightarrow X_k$	-
13	$X_i * X_j \rightarrow X_k$	
14	$X_i / X_j \rightarrow X_k$	$X_j \neq 0$
15	If $X_i \geq c$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	-
16	If $X_i \geq X_j$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	-
17	$\log_e X_i \rightarrow X_k$	$X_i > 0$
18 & 19	Not used	
20	$\sin X_i \rightarrow X_k$	-
21	$\cos X_i \rightarrow X_k$	-
22	$\arctan X_i \rightarrow X_k$	-
23	$X_i^{X_j} \rightarrow X_k$	$X_i > 0$
24	$c^{X_i} \rightarrow X_k$	$c > 0$
	Not defined	
	If $X_j = a_1$ or $a_3 \ldots, a_7$, then $c \rightarrow X_k$; otherwise X_k remains unchanged.	
	If X_i is blank, then $c \rightarrow X_k$; otherwise X_k remains unchanged.	$(X_i \neq -0)$*

*Note that in reading numeric fields, a blank field and -0 are equivalent.

When a violation of a restriction in the right-hand column occurs during transgeneration, the program will print a diagnostic message. Most programs will proceed to the next problem, if any. Some programs will delete the case where the violation occurred and continue the computation. Other programs will screen all the input data for additional restriction violations before proceeding to the next problem.

2.3.3.2 Standard Transgeneration Cards

Standard Transgeneration Cards are used with programs which use Standard Data Input. Let p denote the number of variables in the data matrix and m the maximum number of variables allowed by the program for any problem. Any of the variables $x_1, \ldots, x_m$ may be used in transgeneration. The initial values of the first p variables are read from the input data file (Data Cards or Alternate Input Tape). The initial values of the remaining m-p variables are left over from previous calculations. After transgeneration of a particular case, the values of the first p + q variables for that case are used as the values of the transgenerated variables. If the p + q variables required for the computation are not the first p + q, they must be relocated. This may be done by using transgeneration code 08 with c = 0. The numbers p and q (q may be positive, negative, or zero) are specified on the Problem Card. The indexes i, j, and k from the transgeneration list may exceed p or p + q but must never exceed m.

Card Preparation

Col. 1-6	222222	(Mandatory).
Col. 7-9	Variable index k.	
Col. 10, 11	Code from transgeneration list (restricted by availability in particular program).	
Col. 12-14	Variable index i.	
Col. 15-20	Variable index j or constant c.	
Col. 21-25	Blank.	

Col. 26 Number of a_i's for transformation 40.

Col. 27-32 a_1 value.

Col. 33-38 a_2 value.

...

Col. 63-68 a_7 value.

The constants c, a_1, ..., a_7 are punched with a decimal point if used with variables which have an F-type format and without a decimal point if used with variables which have an I-type format. If there are more than 66 variables, continue on another card of the same form, until p+q variables have been specified.

The variable numbers above refer to variables after transgeneration.

Control Values

1. Delete variable (or dependent variable).
2. Free variable.
3. Low-Level forced variable.

...

9. High-level forced variable.

If no Control-Delete Cards are included, or if a field is left blank on the Control-Delete Cards included in the deck, the value will be assigned if the variable is not the dependent variable and the value 1 assigned if it is the dependent variable.

2.3.4 Computational Procedure

Step 1. The data are read and transgenerated. Let p denote the number of variables after

transgeneration, n the number of cases, and x_{ij} the value of the j^{th} variable, after transgeneration, for the i^{th} case. The means:

$$\bar{x}_i = \frac{1}{m} \sum_{k=1}^{n} x_{ki} \qquad i = 1, \ldots, p$$

are computed and, if called for, printed. If a zero regression intercept is not requested on the Problem Card, the matrix A:

$$a_{ij} = \sum_{k=1}^{n} (x_{ki} - \bar{x}_j) \qquad i, j = 1, \ldots, p$$

is computed. If a zero regression intercept is requested, the matrix A:

$$a_{ij} = \sum^{n} x_{ki} x_{kj} \qquad i, j = 1, \ldots, p$$

is computed instead.

Step 2. The The covariances, standard deviations, and correlations:

$$s_{ij} = a_{ij}/(n-1) \qquad i, j = 1, \ldots, p$$

$$s_i = \sqrt{s_{ii}} \qquad i = 1, \ldots, p$$

$$r_{ij} = s_{ij}/s_i s_j \qquad i, j = 1, \ldots, p$$

are computed and, if called for, printed. It should be noted that if the zero regression intercept option is chosen, these statistics will not be centered about the mean. A similar statement applies to all the computations which follow.

Step 3. At each step in the stepwise regression procedure, the variables $x_1, \ldots, x_p$ are divided into two disjoint sets: $x_{i_1}, \ldots, x_{i_q}$: the independent variables in the regression equation
$x_{j_1}, \ldots, x_{j_r}$: the remaining variables including the dependent variable $y = x_d$.

For purposes of exposition, assume that $x_{i_1}, \ldots, x_{i_q}$ equation at a typical step then has the form:

$$y = \alpha + \beta_1 x_1 + \quad + \beta_p x_p + e.$$

Let

$$A = \begin{matrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{matrix} \text{ where } A_{11} \text{ is } q \times q$$

be a partition of the matrix A from Step 1; let

$$B = \begin{matrix} A_{11}^{-1} & A_{11}^{-1} A_{12} \\ A_{21} A_{11}^{-1} & A_{22} - A_{21} A_{11}^{-1} A_{12} \end{matrix}$$

let $m = n-1$ if a zero regression intercept is not requested; and let $m = n$ if it is requested. For each step in the stepwise procedure, the following are computed and printed:

1. The residual degrees of freedom, sum of squares, and mean square:

   $df = m-q$, $SS = b_{dd}$, $MS = SS/df$

2. The regression degrees of freedom, sum of squares, mean square, and F value:

   $rdf = q$ $\qquad$ $RSS = a_{dd} - b_{dd}$

   $RMS = RSS/rfd$ $\qquad$ $F = RMS/MS$

3. The standard error of estimate and multiple correlation coefficient:

$$S = \sqrt{MS} \qquad R = \sqrt{RSS/a_{dd}}.$$

For each independent variable x_i in the regression equation, the following are computed and printed:

1. The regression coefficient, its standard error, and F value:

$$\beta_i = b_{id},\ S_i = b_{ii}^{1/2} S,\ F_i = (\beta_i/S_i)^2$$

2. If a zero regression intercept is not requested, the intercept α is computed:

$$\alpha = \sum_{i=1}^{q} \beta_i \bar{x}_i .$$

For each independent variable x_i not in the regression equation, the following are computed and printed:

The tolerance level, partial correlation, coefficient, and F value:

$$T_i = b_{ii}/a_{ii},\ R_i = \frac{b_{id}}{\sqrt{b_{ii} b_{dd}}},\ F_i = \frac{b_{id}^2 (m-2-1)}{b_{ii}\, b_{dd} - b_{id}^2}$$

Step 4. To move from one step to the next, an independent variable is added to or removed from the regression equation according to the following three rules:

1. If there are one or more independent variables in the regression equation whose control value, as specified by the Control-Delete Card, is 2 (i.e., a free variable) and whose F value is less than the "F-to-remove" value specified on the Subproblem Card, the one with the smallest F value will be removed.

2. If no variable is removed by item 1 above and there are one or more independent variables not in the regression equation which pass the tolerance test and have control values of 3 or more (i.e., forced variables), the one which has the highest control value and the highest F value among all with the same control value will be added. An independent variable x_i not in the regression equation is said to pass the tolerance test if its tolerance value T_i is greater than or equal to the "minimum tolerance value" specified on the Subproblem Card.

3. If no variable is removed by item 1 or added by item 2 and there are one or more independent variables not in the regression equation which pass the tolerance test, have a control value of 2 (i.e., a free variable), and an F value greater than or equal to the "F-to-enter" value specified on the Subproblem Card, the one with the highest F value will be added. If no variable is added or removed by items 1, 2, or 3, the stepwise procedure is terminated.

Step 5. If a list of residuals is called for on the Subproblem Card, the residuals:

$$r_i = y_i - \alpha - \sum_j \beta_j x_{ij} \qquad i = 1, \ldots, n$$

are computed and printed. The summation is over all indexes j of independent variables x_j in the regression equation at the last step.

2.3.5 Machine Procedure

The Computational Procedure section (2.3.4) explains in terms of the input data what is computed; however, it offers few details on how the machine accomplishes the computation. The purpose of this section is to present more detailed information about the machine procedure. It is written in the imperative mood to conform with the commands presented by the program to the machine.

2.3.5.1 Main Program

1. Read a Problem Card. Read the data and transgenerate. Let

$$x_{ij};\ i = 1, \ldots n;\ j = 1, \ldots,\ p$$

denote the i^{th} case value of the j^{th} variable after transgeneration. If a regression intercept is called for, compute a matrix A and vector B as follows. Set:

$$a_{ij} = 0 \qquad 1 \leq i \leq j \leq p$$

$$b_i = x_{1i} \qquad i = 1, \ldots, p$$

and apply the recursion relations:

$$b_i = b_i + x_{ki} \text{ followed by}$$

$$a_{ij} = a_{ij} + \frac{1}{k-1}\left[x_{ki}\,(kx_{kj} - b_j)b_i\left(\frac{b_j}{k} - x_{kj}\right)\right]$$

for k = 2, . . . , n. Set m = n-1. If a regression intercept is not called for, set:

$$a_{ij} = b_i = 0 \quad 1 \leq i \leq j \leq p$$

and apply the recursion relations for k = 1, . . . , n. Set m = n. Save the transgenerated data on tape.

2. Replace B and A by a mean vector and covariance matrix:

$$b_i = b_i/n \qquad i = 1, \ldots, p$$

$$a_{ij} = \frac{1}{n-1} a_{ij} \qquad 1 \leq i \leq j \leq p$$

and compute the standard deviations:

$$s_i = \sqrt{a_{ii}} \qquad i = 1, \ldots, p$$

If called for, print the means, standard deviations, and covariance matrix.

3. Replace A by a correlation matrix:

$$a_{ij} = a_{ij}/s_i s_j \qquad i, j=1, \ldots, p$$

and print if called for.

4. Read a Subproblem Card and restore the matrix A:

$$a_{ij} = a_{ji} \qquad 1 \leq j \leq i \leq p$$

$$a_{ii} = 1 \qquad i = 1, \ldots, p$$

Set up the control vector C:

$$c_i = \text{control value for the } i^{th} \text{ variable}$$

and set the number of variables in the equation q=0.

5. Call subroutine STEPRG. This subroutine will step A, C, and q, and send back an index k and a flag.

-1 variable k removed

Flag = 0 go to Step 9

1 variable k added

6. Let d denote the index of the dependent variable. Compute and print:

   a. $q = q + \text{Flag}$

   b. The residual degrees of freedom, sum of squares, and mean square:

   $$df = m-q, \quad SS = m\, s_d^2\, a_{dd}, \quad MS = SS/df$$

   c. The regression degrees of freedom, sum of squares, mean square, and F value:

   $$rdf = q \qquad RSS = s_d^2 - SS$$

   $$RMS = RSS/df \qquad F = RMS/MS$$

   d. The standard error of estimate and the multiple correlation coefficient:

   $$S = \sqrt{MS}, \qquad R = \sqrt{1-a_{dd}}$$

7. A variable with index i is in the regression of $c_i \leq 0$. For each such variable, compute and print the regression coefficient, its standard error, and F value:

$$\beta_i = \frac{s_d\, a_{id}}{s_i}, \quad S_i = \frac{S}{s_i}\frac{a_{ii}}{m}, \quad F_i = (\beta_i/S_i)^2.$$

If the regression intercept α is called for, compute and print:

$$\alpha = b_d - \sum_i \beta_i\, b_i$$

where the summation is over all i such that $c_i \leq 0$.

8. A variable with index $i \neq d$ is not in the regression if $c_i \geq 1$. For all such variables, compute and print the tolerance level, partial correlation coefficient, and F value:

$$T_i = a_{ii}, \quad R_i = a_{id} / \sqrt{a_{ii} a_{dd}}$$

$$F_i = a_{id}^2 (m-q-1)/(a_{ii} a_{dd} - a_{id}^2).$$

Go to Step 5.

9. If called for, compute and print the residuals:

$$r_i = \alpha + x_{id} - \sum_j \beta_j x_{ij} \qquad i = 1, \ldots, n$$

where the summation is over all j such that $c_j \leq 0$.

10. Return to Step 4 and continue reading Subproblem Cards until there are no more.

11. Return to Step 1 and continue reading Problem Cards until there are no more.

2.3.5.2 Subroutine STEPRG

1. Compute

$$V_{in} = \frac{F_{in} \cdot a_{dd}}{F_{in} + m-q-1 + 2}$$

$$V_{out} = \frac{F_{out} \cdot a_{dd}}{m-q} \quad -7$$

where F_{in} and F_{out} are the F values to enter and remove.

2. Find the k such that $c_k < 1$ and k minimizes:

$$V_k = c_k \frac{a_{kd}^2}{a_{kk}}$$

If $V_k < V_{out}$, set Flag = -1, increase c_k by 9, and go to Step 4.

If $V_i \geq V_{out}$ or $c_k \geq 1$ for all k, go to Step 3.

3. Let Tol be the minimum tolerance. Find the k such that $c_k > 1$, $a_{kk} \geq$ Tol, and k maximizes:

$$V_k = c_k + \frac{a_{kd}^2}{a_{kk}} .$$

If $V_k \geq V_{in}$, set Flag = 1, reduce c_k by 9, and go to Step 4.

If $V_k < {}_{in}$ or there is no k such that $c_k > 1$ and $a_{kk} \geq$ Tol, set Flag = 0 and go to Step 5.

4. Call subroutine STEP and return

5. Set m = m + Flag and return.

2.3.5.3 Subroutine STEP

1. Set:

$$u_i = \begin{cases} a_{ik} & 1 \leq i < k \\ -\text{Flag} & i = k \\ a_{ki} & k < i \leq p \end{cases}$$

$$a_{ik} = a_{kj} = 0 \qquad i \leq k < j$$

2. Step the upper diagonal part of A:

$$a_{ij} = a_{ij} - \frac{u_i u_j}{a_{kk}} \qquad 1 \leq i \leq j \leq p+1$$

and return.

2.4 OMCC PROGRAM

2.4.1 Purpose

The Ozone Monitor Coefficient Converter (OMCC) program takes the output from the Regression Analysis program (EPASTREG) and converts the selected coefficients into double-precision octal form for insertion via punched paper tape into the PDP-8/L computer.

2.4.2 Control Cards: Fortran Unit 5 (three cards required)

1. NOC, NOE: Format (8G10.0) - first card

   NOC - Number of data channels

   NOE - Number of equations = 2 (normally) for Ozone Monitor

2. INPD, NOS, SN, SCALE, NFSA; Format (4G10.0, 6X, A4) - next two cards

   INPD - Input Device

   NOS - Number of steps in the regression analysis (set to 1 for first pass; then set to total number of steps)

   SN - Regression step number of interest (set to 0 for first pass; then to desired step number)

   SCALE - "Magic number" for converting decimal floating point numbers to double-precision octal number for insertion into PDP-8/L
   = 8388.608 for equation 1 (starting address, 4200)
   = 2048. for subsequent equations (starting address, 4400)

   NFSA - Number for starting address in PDP-8/L for data

2.4.3 Data Input: Fortran Unit 5 (read EPASTREG punched output cards)

1. NSTP, NCOEFF, SE; Format (2I10, F10.0) - one card

   NSTP - The step number in regression analysis

   NCOEFF - Number (how many) of non-zero coefficients in the step

   SE - Standard estimate of error

2. ICN, COEF(ICN); Format (4(I10, F10.5)) - one or more cards

ICN - Index of channel number

COEF(ICN) - Coefficient for ICN channel

(Note 1 and 2 repeat for NOS times)

2.4.4 Output Data File: Fortran Unit 7 (card punch output for card-to-tape PDP-8/L input)

LF, NFSA, N1 - N4, CR, LF, N5 - N8, CR;
Format (A1, A4, 1H/, 4I1, A1, A1, 4I1, A1)

LF - Line feed code (Z69)

NFSA - Starting address in PDP-8/L for data

N1 - N8 - Double-precision octal coefficients

CR - Carriage return code (Z7A)

2.4.5 Printer: Fortran Unit 6

2.5 EPAMSTDV PROGRAM

2.5.1 Purpose

This program computes mean and standard deviations for up to 50 variables and any number of cases.

2.5.2 Data Input Cards: Fortran Unit 5

1. NOV, NOC, NPRNT: Format (2I4, 1X, I1)

   NOV - Number of variables

   NOC - Number of cases

   NPRNT - 1 - Print input data

2. DATA (I), I=1, NOV; Format (6F10.5)

2.5.3 Printer: Fortran Unit 6

2.6 EPA PLOTR PROGRAM

2.6.1 Purpose

This program plots raw scan data obtained from the Ozone Monitor Instruments. The program prints the channel number, fitter wheel position number, and wavelength. In addition, it prints the case number and the raw data values.

2.6.2 Data Input Cards: Fortran Unit 5

1. BCD - Title of the plot, Format (20A4)
2. N - Total number of plots requested, Format (I10)
3. NSTEP - Step number of each plot, Format (I10) (one per card)

2.6.3 Data Input Cards: Fortran Unit 9

1. N - Number of cases, Format (I10)
2. NOC, FWPL (I), I = 1, NOC; Format (8I10)

   NOC - Number of channels

   FWPL - Filter wheel position list
3. RAD (I), I = 1, NOC, ID; Format (20A4)

   RAD (I) - Radiant reading

   ID - Identification number

2.6.4 Printer: Fortran Unit 6

2.7 FACTORAN PROGRAM

This factor analysis program is from the Scientific Subroutine Package produced by IBM (Ref. 5) and modified by Bendix Aerospace Systems Division.

2.7.1 General Description

This program performs a factor analysis of up to 198 input variables. The factoring may be done by using either covariance or correlation matrices. Three types of rotation are available, all based on the oblimin criteria. In the first, the factors are restricted to be orthogonal which yields, among others, quartimax and varimax rotations. In the second, the criteria are applied to the reference factor structure and the factors are allowed to be oblique which yields the standard oblimin rotations. In the third, the criteria are applied to the primary factor loadings, allowing the factors to be oblique and yielding simple loading rotations (Ref. 2). Factor scores may be estimated. Data input may be in the form of raw data, a correlation matrix, a covariance matrix, or a factor loading matrix. The program leaves communalities unaltered.

The output consists of:

- Means and standard deviations.
- Correlation or covariance matrix.
- Eigenvalues and cumulative proportion of total variance.
- Factor loading matrix before rotation.
- Rotated factor loading matrix.
- Correlation matrix of the rotated factors.
- Factor scores.

2.7.2 Job Control Setup

The program must be run in a large partition of approximately 270K size. The program requires two temporary disk files with data set reference numbers 1 and 2.

FACTORAN has been equipped with a flexible option input scheme. Option cards may be input in any order, with four exceptions:

- Format card(s) must follow DATAFMT, FSFMT, FACTFMT option cards.
- Label for data set must follow LABEL option card.
- EXEC card must be the last in definition of a set of options for one problem; i.e., it triggers the analysis.
- Any data input from the card reader must follow the EXEC card belonging to its problem definition.

Any number of problems may be submitted in one run. If more than one problem is submitted at once, all options chosen for the first hold for subsequent problems, unless explicitly changed with new option cards.

Any option card which has a value associated with it must have that value punched in columns 9-20 with decimal point; i.e., the card is read on format (A8, F12.0).

2.7.3 Data Option Cards

The data input and output options available are described below. A "b" indicates a blank column.

CODE		DEFINITION
DATA	default	Input will be raw data only.
FACTOR		Means, standard deviations, number of factors, eigenvalues, unitary transform matrix, and factor matrix will be read in. Raw data must still be input as well. (This option allows the user to use punched output from a previous run as input to avoid duplication of these calculations.) The program automatically punches these values with each run made on the DATA option. They are output in the order listed above, followed by the factor score coefficient matrix for non-standard data. Formats for these are:

		Means: 4F18.8 Std dev: 4F18.8 No. fact: 15 Eigenvalues: 5F15.8 (No. fact of them) Unitary transform: 5F15.8 (Row wise) Factor Matrix 5F15.8 (Column wise)
SIGDATA	default	Data input will be in signature data processing form, KT =1 or KT = 2.
DATAFMTbxx.		Data input will be in format following this card. xx indicates how many format cards follow. xx defaults to 1. The maximum is 9 cards.
SIGFSCOR	default	Factor Score output will be in signature data processing form. KT = 5.
FSFMTbbbxx.		Factor Score output will be in format following this card. xx indicates how many format cards follow. xx defaults to 1, maximum is 9. The file will be preceded by a record written on format 3I10 giving number of observations, number of variables, number of factors.
FACTFMTbxx.		Factor matrix input will be in format following this option card. xx indicates how many format cards follow. xx defaults to 1. The maximum is 9 cards. This format applies only to the factor matrix and not the means, standard deviations, eigenvalues, or unitary transform matrix of the FACTOR option.
DATAUNITxx.		Data input unit. xx defaults to 8.
FSUNITbbxx.		Factor score output unit. xx defaults to 9.

FACTUNITxx.		Factor matrix input unit (including means, standard deviations, number of factors, eigenvalues, unitary transform matrix). xx defaults to 5.
INGROUPbxx.		Group number of group from input file to be analyzed. This applies to signature data processing form of input. Specify as many of these cards in one problem (in ascending order of group number) as number of groups desired in factor score output file.
LABEL		Card following this contains an 8-character code to be used in the label field of newly created signature data processing type file; i.e., use when input is not SDPS, but want factor score output to be. The label defaults to blanks.
NUMOBSbbxxxxx		Number of observations in raw data input. If SIGDATA, defaults to read off input file. If DATAFMT and not from card reader, defaults to read off initial record of input file--the first variable on 2I10 format. Otherwise, defaults to 90.
NUMVARbbxx.		Number of variables in raw data input. If SIGDATA, defaults to read off input file. Otherwise, defaults to 8.
NTALPRNT	default	Do not print all matrices before rotation.
ALLPRNT		Print all matrices before and after rotation.

As an example of the options available, consider the following setup as one run with five problems.

	INGROUP 1. INGROUP 2. INGROUP 3. EXEC	Input from default unit in SDPS format. Three output groups in one file on default unit in SDPS format.
Once you have specified an EXEC card, a new output unit must be specified for subsequent problems in the same run. You can use the same input file in more than one problem.	FSUNIT 11. INGROUP 4. EXEC FSUNIT 12. INGROUP 5. EXEC FSUNIT 13. INGROUP 6. EXEC	Three output files on separate units, one group each. Input is from same file as first problem since did not specify DATAUNIT.
	DATAUNIT 10. FSUNIT 14. INGROUP 1. EXEC	Input from unit 10 in SDPS format. Output unit 14 will have one output group in SDPS format.

2.7.4 Analysis Option Cards

The types of data analysis options available are described below.

CODE		DESCRIPTION
MEAN	default	Correlation or covariance to be done about the mean.
ORIGIN		Correlation or covariance to be done about the origin.
CORREL	default	Correlation matrix to be used in analysis.
COVAR		Covariance matrix to be used in analysis.
NOKAIS	default	Do not Kaiser normalize before rotation.
KAIS		Kaiser normalize before rotation.
NOROT	default	No rotation.
ORTHROT		Orthogonal rotation.

OBROT		Oblimin rotation.
SIMPLOAD		Oblique rotation for simple loadings.
EIGMINbbxxxxx	default	Minimum eigenvalue xxxxx defaults to 0.0.
FACTKEEPxxxxx		Number of factors to keep. xxxxx has no default.
PCNTKEEPxxxxx		Maximum percentage of total variance to be explained. xxxxx has no default.
GAMMAbbbxxxxx		Variable gamma for rotation. xxxxx defaults to 1.0 if orthogonal rotation. xxxxx defaults to 0.5 if oblimin rotation. xxxxx defaults to 0.0 if oblique rotation for simple loadings

Some values of gamma which lead to standard rotations:

	$\gamma = 0.0$	$\gamma = 0.5$	$\gamma = 1.0$
ORTHOGONAL	quartimax		varimax
OBLIMIN	quartimin	bi-quartimin	covarimin
SIMPLE LOADINGS	direct quartimin		

2.8 FACREC PROGRAM

2.8.1 Purpose

This program reconstructs spectral data by recombination of the factor analysis of the data. By retaining only the more prominent factors, the reconstructed data accurately reproduce the spectral patterns present, while eliminating the effects of the instrument noise which exhibits no channel-to-channel correlation.

2.8.2 Method

The program accepts as input the factor matrix A and a list of factors score vectors X. The reconstructed channel reading vectors y are given by:

$$y = Ax.$$

At the user's option, the program will plot either the individual factors against wavelength or the factor scores in order of occurrence, or both.

2.8.3 Option Cards: Fortran Unit 5: Format (1A8)

1. PRRD; Causes the reconstructed data to be printed
2. PFM; Causes the factor matrix to be plotted
3. PLOTFS; Causes the factor scores to be plotted
4. EXEC: Causes execution to begin under control of options already specified
5. CWL(I), I = 1, NV; Format (8F10.0)
   CWL = Center wavelength
6. FINISH; Terminates execution.

2.8.4 Data Input File: Fortran Unit 8

1. NC, NV, NF; Format (3I10)

   NC = Number of cases

   NV = Number of variables

   NF = Number of factors

2. A (I, J), I = 1, NV,

J = 1, NF; Format (20A4)

A (I, J) = Factor matrix element for the Ith variable and Jth factor

3. X (J), J = 1, NF; Format (20A4)

X (J) = Score for Jth factor.

2.8.5 Output Data File: Fortran Unit 9

1. NC; Format (I10)

NC = Number of cases

2. y (I), I = 1, NV; Format (20A4)

y (I) = Reconstructed reading for Ith channel.

2.8.6 Printer: Fortran Unit 6

2.9 DIMAT PROGRAM

2.9.1 Purpose

This program generates a matrix for use in Program DICIS to correct spectroradiometer data for the specific colorations introduced by a particular instrument and to interpolate the data for points lying between center wavelengths of the channels.

2.9.2 Method

A list of filter wheel position numbers is read and a corresponding list of channel center wavelengths is computed by means of a polynomial. The coefficients in the polynomial were previously obtained by regression of manufacturer's data for the particular filter wheel used in an instrument constructed for the Naval Applied Science Laboratory. The filter wheel position list must, of course, be the same as the list selected by the user and employed in the instrument when spectral data were collected. With rare exceptions, the data were collected with 33 channels spaced about 0.125 micron apart and covering a spectral range from 8.5 to 12.5 microns.

A four-point cubic interpolation rule is used to generate the elements a_{IJ} of the interpolation matrix A:

$$a_{I,J-1} = \left[\left(\frac{-a}{\lambda_{J+1} - \lambda_J} + 2 \right) \frac{a}{\lambda_{J+1} - \lambda_J} - 1 \right] \frac{a}{\lambda_{J+1} - \lambda_{J-1}}$$

$$a_{I,J} = \left[\left(\frac{-a}{\lambda_{J+2} - \lambda_J} + \frac{a}{\lambda_{J+1} - \lambda_J} \right) \frac{a}{\lambda_{J+1} - \lambda_J} + \frac{a}{\lambda_{J+2} - \lambda_J} - \frac{3a}{\lambda_{J+1} - \lambda_J} \right] \frac{a}{\lambda_{J+1} - \lambda_J} + 1$$

$$a_{I,J+1} = \left[\left(\frac{-2a}{\lambda_{J+1} - \lambda_J} + \frac{a}{\lambda_{J+1} - \lambda_{J-1}}\right)\frac{a}{\lambda_{J+1} - \lambda_J} - \frac{2a}{\lambda_{J+1} - \lambda_{J-1}} + \frac{3a}{\lambda_{J+1} - \lambda_J}\right]\frac{a}{\lambda_{J+1} - \lambda_J} + \frac{a}{\lambda_{J+1} - \lambda_{J-1}}$$

$$a_{I,J+2} = \frac{a}{\lambda_{J+1} - \lambda_J} - 1\ \frac{a^2}{(\lambda_{J+1} - \lambda_J)(\lambda_{J+2} - \lambda_J)}$$

where

λ_J = center wavelength of Jth channel $\quad 1 \leq J \leq 33$

λ_I = Ith wavelength of interpolation $\quad 1 \leq I \leq 251$

$a = \lambda_I - \lambda_J$

and the channel subscript J is altered as required to maintain the relation

$$\lambda_J \leq \lambda_I \leq \lambda_{J+1}.$$

Next, the filter wheel transmission matrix B is computed. The shape of the spectral transmission of each channel is approximately that of a Gaussian distribution, and the expression used is:

$$b_{J,I} = \frac{T_{J,I}}{\sum_{I=1}^{251} T_{J,I}}.$$

The inverse of the product of the matrixes A and B is then computed and multiplied by A, producing the matrix D:

$$D = A(BA)^{-1}$$

Finally, a responsivity correction factor is computed for each channel. Experimental data were obtained by operating the instrument in an ambient temperature of 297.0°K and exposing it to the radiation from a blackbody tank at 306.5°K. Thirty spectra were collected and averaged, and a correction factor for each channel was obtained by dividing the known spectral radiance difference at the wavelength for each channel by the average reading for each channel. The correction factors are then applied to each column of the matrix D, completing the required calculations.

2.9.3 Data Input Cards: Fortran Unit 5

1. NOC, X(I), I = 1, NOC; Format (8F10.0)

   NOC = number of channels

   X(I) = filter wheel position number for Ith channel

2. BW: Format (F10.0)

   BW = channel half-power bandwidth in microns

3. TR(J), J = 1, NOC; Format (8F10.0)

   TR(J) = average reading of Jth channel

2.9.4 Data Output File: Fortran Unit 10

1. NOC; Format (I10)

2. A(I, J), I = 1, 251; J = 1, NOC; Format (20A4)

   A(I, J) = element of the deconvolution and interpolation matrix for the Ith wavelength and the Jth channel.

2.9.5 Printer: Fortran Unit 6

2.10 ALFMAT PROGRAM

2.10.1 Purpose

This program generates a matrix of absorption coefficients and writes it into a data file for use by Program DICIS. The input data are from cards, with each contaminant represented requiring 131 readings covering the spectral range from 7.5 to 14.0 microns in steps of 0.05 micron. Linear interpolation is used to provide higher nominal resolution, and the spectral range is restricted for the convenience of the subsequent computation by Program DICIS. Each contaminant is represented in the output by 251 readings covering the spectral range from 8.0 to 13.0 microns in steps of 0.02 micron.

2.10.2 Control Cards: Fortran Unit 5

NOCT; Format (I10)

NOCT = number of contaminants

2.10.3 Data Input Cards: Fortran Unit 5

X(I), I = 1, 131; Format (10X, 7F10.0)

X(I) = absorption coefficients

2.10.4 Data Output File: Fortran Unit 11

1. CONNAM(I), I = 1, 10; Format (10A8)

   CONNAM(I) = name or mnemonic for (I)th contaminant

2. A(I, J), I = 1, 251, J = 1, 10; Format (20A4)

   A(I, J) = (I)th absorption coefficient for (J)th contaminant

2.10.5 Printer: Fortran Unit 6

2.11 FILTER PROGRAM

2.11.1 Purpose

This program accepts spectral filter and channel allocation data from card input and outputs lists of individual channel parameters to a data file for use by Program DICIS.

2.11.2 Control Cards: Fortran Unit 5

NOFT; Format (I10)

NOFT = number of filter types

2.11.3 Data Cards: Fortran Unit 5

1. NF(k), BWL(k), DWL(k),

   k = 1, NOFT; Format (I10, 2F10.0)

   NF(k) = number of channels to be assigned to filter type K

   BWL(k) = beginning wavelength for the channels of type K

   KWL(k) = difference between center wavelengths for channels of type K

2. FTR(I, k), I = 1, L, k = 1, NOFT; Format (3x, F9.3)

   FTR(I, k) = filter transmission at the (I-1)th wavelength deviation from the center wavelength of filter type K

   L = value of I for which first zero transmission is read

2.11.4 Output Data File: Fortran Unit 12

1. NOFT, NOC; Format (2 I 10)

   NOFT = number of filter types

   NOC = total number of channels summed over all filter types

2. IFT(j), FWL(j), PTR(j),

   j = 1, NOC; Format (I10, 2F10.5)

   IFT(j) = filter type number for jth channel

   FWL(j) = center wavelength of jth channel

   PTR(j) = peak transmission of jth channel

3. FTR(I, k), I = 1, 251, K = 1, NOFT; Format (8F10.6)

2.11.5 Printer: Fortran Unit 6

2.12 DICIS PROGRAM

2.12.1 Purpose

This program processes spectral data in three major steps. In the first step, the input data are deconvolved and interpolated to provide a high-resolution representation of a radiance difference spectrum corrected for the known properties of the collecting instrument. In the second step, the spectra are contaminated by the absorption spectra of various user-chosen atmospheric contaminants. In the third step, an instrument simulation is performed to convert the final radiance spectra to a set of channel readings consistent with the user-specified number of channels, channel bandwidths, center wavelengths, and instrument noise.

2.12.2 Method

Three arrays of constants are read from data files created by other programs. The first array consists of the elements of a deconvolution and interpolation matrix generated by Program DIMAT. The second array consists of the spectral absorption coefficients for up to 10 atmospheric contaminants produced by Program ALFMAT. The third array consists of a number of channel description parameters provided by Program FILTER.

After various user options are determined by reading a series of control cards, tables of spectral transmittance are prepared for each of the contaminants selected:

$$T_{I,k} = \exp(-\alpha_{I,k} \cdot CL_k)$$

where $T_{I,k}$ is the transmittance of the kth contaminant at the Ith wavelength, $\alpha_{I,k}$ is the corresponding absorption coefficient, and the CL product is chosen to achieve a specific peak absorption.

The execution of the main computation loop begins with the reading of the first input background spectrum and is repeated for each additional spectrum to be processed. A refined version of the background spectrum is obtained by the relation:

$$y_I = \sum_{j=1}^{33} A_{I,j} = X_j, \quad 1 \leq I \leq 251$$

where X_j is the input reading of the jth channel, y_I is the corrected spectral radiance at the Ith wavelength, and A_{Ij} is a 251 by 33 deconvolution and interpolation matrix.

The effects of an airborne contaminant may be introduced in the spectrum by use of the relation:

$$X_I = y_I \cdot T_{I,k}$$

under the restriction that the temperature of the contaminant is the same as that of the instrument ambient temperature.

The instrument simulation portion of the program generates a set of channel readings by summing the products of the spectral radiance and the individual spectral channel transmissions. The user-specified amount of random noise is added to each channel reading, and a target value for the subsequent regression analysis is appended to the set of channel readings.

The target value is set to a non-zero value only for those backgrounds which have been modified by a target contaminant. It is computed by obtaining the vector difference between the original and the modified spectral radiance vector and finding the length of the component perpendicular to the original vector. The sign is taken positive or negative according to whether absorption or emission effects prevailed.

2.12.3 Option Cards: Fortran Unit 5

1. PRIN; causes input spectral data to be printed
2. PROUT; causes refined spectral data to be printed
3. PRCR; causes output channel readings to be printed
4. REWIN; causes Fortran Unit 8 to be rewound before processing continues

5. RNOISE; causes another card to be read in Format (F10.0) which specifies random noise level to be used

6. BBSYN; causes another card to be read in Format (I10) which specifies the number of blackbody spectra to be synthesized and included with the refined spectra for processing

7. TARGET; causes sufficient additional cards to be read in Format (I10, /4 (I10, F10.0)) to specify the number of target contaminants to be used, their identity, and peak absorption

8. NONTAR; causes sufficient additional cards to be read in Format (I10, /4 (I10, F10.0)) to specify the number of non-target contaminants to be used, their identity, and peak absorption

9. PLOTBS; causes sufficient additional cards to be read in Format (I10, /(8I10)) to specify the number of background spectra to be plotted and to identify the particular background for which plots are desired

10. PLOTCB; causes sufficient additional cards to be read in Format (I10, /(8I10)) to specify the number of contaminated spectra to be plotted and to identify the particular backgrounds and contaminants for which plots are desired

11. EXEC; causes execution to begin under control of options already specified

12. FINISH; terminates execution.

2.12.4 Input Data File: Fortran Unit 10

1. NOMC; Format (I10)

   NOMC = Number of matrix channels

2. A (I, J), I = 1, 251

   J = 1, NOMC; Format (20A4)

   A (I, J) = Element of deconvolution and interpolation matrix for Ith wavelength and Jth channel.

2.12.5 Input Data File: Fortran Unit 11

1. CONNAM (K), K = 1, 10; Format (10A8)

   CONNAM (K) = Name of Kth contaminant

2. ALF (I, K), I = 1, 251

   K = 1, 10; Format (20A4)

   ALF (I, K) = Absorption coefficient of Kth contaminant at Ith wavelength.

2.12.6 Input Data File: Fortran Unit 12

1. NOFT, NOOC; Format (2I10)

   NOFT = Number of filter types

   NOOC = Number of output channels

2. IFT (J), FWL (J), PTR (J),

J = 1, NOOC; Format (I10, 2F10.0)

IFT (J) = Index of filter type for Jth output channel

FWL (J) = Filter center wavelength for Jth output channel

PTR (J) = Peak transmission of Jth output channel.

2.12.7 Input Data File: Fortran Unit 8

1. NCI; Format (I10)

NCI = Number of input spectra

2. X (J), J = 1, 33; Format (20A4)

X (J) = Jth channel reading.

2.12.8 Output Data File: Fortran Unit 9

1. NCO; Format (I10)

NCO = Number of output cases

2. X(J), J = 1, NOOC, TARG; Format (6F10.3)

X(J) = Output reading for Jth channel

TARG = Target value associated with current readings.

2.12.9 Printer: Fortran Unit 6

2.13 PLTSMF PROGRAM

2.13.1 Purpose

This program plots spectral modulation functions from regression analysis data and channel filter response data.

2.13.2 Method

The spectral modulation function is obtained by summing the products of the spectral response of each channel with its associated regression coefficient. The summation is performed for each wavelength of interest using the relation

$$r_I = \sum_{J=1}^{N} C_J \cdot T_{J,I}$$

where r_I is the response at the Ith wavelength, C_J is the coefficient for the Jth channel, $T_{J,I}$ is the transmission of the Jth channel at the Ith wavelength, and N is the number of channels. The values of r_I are plotted against the wavelength corresponding to the index I.

2.13.3 Data Input Cards: Fortran Unit 5

NSTBP,

NSI (I), I = 1, NSTBP; Format (8110)

NSTBP = Number of regression steps to be plotted

NSI (I) = Step number of Ith plot.

2.13.4 Data Input File: Fortran Unit 11

1. NOFT, NOC; Format (2I10)

   NOFT - Number of filter types

   NOC = Number of channels

2. IFT (J), FWL (J), PTR (J),

J = 1, NOC Format (I10, 2F10.5)

IFT (J) = Filter type number for Jth channel

FWL (J) = Center wavelength of Jth channel

PTR (J) = Peak transmission of Jth channel

3. FTR (I, K), I = 1, 251

K = 1, NOFT; Format (8F10.6)

FTR (I, K) = Filter transmission at the (I-1)th wavelength deviation from the center wavelength of filter type K.

2.13.5 Data Input File: Fortran Unit 12

1. NSTEP, NOCF, SE Format (2I10, F10.5)

NSTEP = Regression step number

NOCF = Number of coefficients in this step

SE = Standard error of estimate for this step

2. NCN (M), COEF (M),

M = 1, NOCF; Format (4(I10, F10.5))

NCN (M) = Channel number of Mth channel included in regression step

COEF (M) = Coefficient for Mth channel included.

2.13.6 Printer: Fortran Unit 6

SECTION 3

REFERENCES

1. M. A. Efroymson, "Multiple Regression Analysis," Mathematical Methods for Digital Computers, Part V, (17), Edited by A. Ralston and H. S. Wilf, John Wiley, New York (1960).

2. R. I. Jennrich and P. F. Sampson, "Rotation for Simple Loadings," Psychometrika (1966).

3. J. H. Wilkinson, "The Calculation of Eigenvectors of Codiagonal Matrices," Computer J., 1, 90-96 (1958).

4. Harry H. Harman, Modern Factor Analysis, Chicago: Univ. Chicago Press (1960).

5. IBM Corporation, System/360 Scientific Subroutine Package (360A-CM-03X) Version III Programmer's Manual, Ref. H20-0205-3, Fourth Edition.